UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,894	10/22/2001	Ching-Jer Liang	JCLA7410	6847

23900 7590 03/15/2005

J C PATENTS, INC.
4 VENTURE, SUITE 250
IRVINE, CA 92618

EXAMINER	
THANGAVELU, KANDASAMY	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 03/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/037,894	LIANG, CHING-JER	
	Examiner	Art Unit	
	Kandasamy Thangavelu	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 October 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 October 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☒ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-11 of the application have been examined.

Foreign Priority

2. Acknowledgment is made of applicant's claim for foreign priority based on applicant's foreign priority application 90116947 dated July 11, 2001 filed in Taiwan. Certified copies of the priority documents have not been submitted by the applicant under 35 U.S.C. 119(a)-(d).

Drawings

3. The drawings submitted on October 22, 2001 are objected to:

In Fig.3, Block 316, "microprocessor jumps from normal operating mode into circuit mode into circuit emulation mode" appears to be incorrect and it appears that it should be "microprocessor jumps from normal operating mode into circuit emulation mode".

Decision block 314 should have only a Yes and a No branches.

The program flow from Block 317 to 302 is incorrect; it should flow from block 318 to 304. The program flow from Block 322 to 308 is incorrect; it should be from Block 322 to Block 304.

In Block 320 "counter value reach an upper value" appears to be incorrect and it appears that it should be "counter value reaches an upper value".

The order of Blocks 310 and 312 is incorrect. The order shows that the instruction counter and the cycle counter are initialized after executing the program. The counters should be initialized before executing the program. If initialization involves inputting initial values, then it should be done in emulation mode and not in normal operating mode.

Appropriate corrections are required.

Abstract

4. The abstract is objected to because of the following informalities:

Lines 4-6, "The instruction counter and the cycle counter is rest to zero. Assessment points are set up along a series of instruction whose operating speed needs to be determined" appears to be incorrect and it appears that it should be "The instruction counter and the cycle counter are rest to zero. Assessment points are set up at the start and end of a series of instructions whose operating speed needs to be determined".

Lines 11-14, "the microprocessor jumps from the normal operating mode back into circuit emulation mode. Microprocessor performance is evaluated by dividing the value inside the cycle counter by the value inside the instruction counter" appears to be incorrect and it appears that it should be "the microprocessor jumps from the normal operating mode back into circuit emulation mode. The values inside the instruction

counter and the cycle counter are read. Microprocessor performance is evaluated by dividing the value inside the cycle counter by the value inside the instruction counter".

Lines 16-18, "After evaluating the microprocessor performance, the microprocessor jumps into the circuit emulation mode again until all instructions are executed or another assessment point is encountered" appears to be incorrect and it appears that it should be "After evaluating the microprocessor performance, the microprocessor jumps into the normal operating mode again until all instructions are executed or another assessment point is encountered".

Appropriate corrections are required.

Specification

5. The disclosure is objected to because of the following informalities:

Specification Page 1, Line 23, "so that any problem area can be deal with appropriately" appears to be incorrect and it appears that it should be "so that any problem area can be dealt with appropriately".

Specification Page 2, Lines 5-6, "An instruction counter and a cycle counter is reset to zero" appears to be incorrect and it appears that it should be "An instruction counter and a cycle counter are reset to zero".

Specification Page 2, Lines 9-11, "to read off the value in the instruction counter and the cycle counter and then produced a report on the performance of the execution"

appears to be incorrect and it appears that it should be "to read off the values in the instruction counter and the cycle counter and then produce a report on the performance of the execution".

Specification Page 2, Lines 14-16, "the microprocessor is triggered into a circuit emulation mode and read off the value in the instruction counter and the cycle counter and produces a report on the performance of the execution" appears to be incorrect and it appears that it should be "the microprocessor is triggered into the circuit emulation mode and the values in the instruction counter and the cycle counter are read off and a report on the performance of the execution is produced".

Specification Page 2, Line 23 to Page 3, Line 2, the microprocessor is triggered into the circuit emulation mode and read off the values inside the instruction counter and the cycle counter and produces a performance report" appears to be incorrect and it appears that it should be "the microprocessor is triggered into the circuit emulation mode and the values inside the instruction counter and the cycle counter are read off and a performance report is produced".

Specification Page 3, Lines 21-23, the microprocessor is triggered into a circuit emulation mode and read off the values in the instruction counter and the cycle counter and to produce a performance report" appears to be incorrect and it appears that it should be "the microprocessor is triggered into the circuit emulation mode and the values in the instruction counter and the cycle counter are read off and a performance report is produced".

Specification Page 4, Lines 9-10, the microprocessor is triggered into a circuit emulation mode and read off the value inside the instruction counter and the cycle counter" appears to be incorrect and it appears that it should be "the microprocessor is triggered into the circuit emulation mode and the values inside the instruction counter and the cycle counter are read off".

Specification Page 4, Line 11, "after a timing pulse traverse each cycle" appears to be incorrect and it appears that it should be "after a timing pulse traverses each cycle".

Specification Page 4, Lines 12-14, the microprocessor is triggered into a circuit emulation mode and read off the value inside the instruction counter and the cycle counter" appears to be incorrect and it appears that it should be "the microprocessor is triggered into the circuit emulation mode and the values inside the instruction counter and the cycle counter are read off".

Specification Page 5, Line 24, "an instruction counter and a cycle counter is reset to zero" appears to be incorrect and it appears that it should be "an instruction counter and a cycle counter are reset to zero".

Specification Page 6, Lines 15-16, "a microprocessor, an instruction counter and a cycle counter is used" appears to be incorrect and it appears that it should be "a microprocessor, an instruction counter and a cycle counter are used".

Specification Page 6, Lines 21-23, "a plurality of instructions in a program is executed. In step 312, both the instruction counter and a cycle counter start to count". It

is not clear what causes the counters to start to count, after a plurality of instructions in a program is executed.

Specification Page 6, Line 24, "when a timing pulse is traversed" is not understood. What is meant by "a timing pulse is traversed"?

Specification Page 7, Line 5, "The microprocessor then branches back to step 302" is not shown in Fig. 3. The microprocessor then branches back to step 308 in Fig. 3.

Specification Page 8, Line 1, "microprocessor, an instruction counter and a cycle counter is used" appears to be incorrect and it appears that it should be "microprocessor, an instruction counter and a cycle counter are used".

Specification Page 8, Lines 11-12, "the microprocessor jumps away from the normal operating mode into the circuit emulation mode" appears to be incorrect and it appears that it should be "the microprocessor jumps from the normal operating mode into the circuit emulation mode".

Specification Page 8, Line 17, "In Step 310, execution of the program is initiating" appears to be incorrect and it appears that it should be "In Step 310, execution of the program is initiated". This also differs from Page 8, Lines 6-7, "In Step 310, a plurality of instructions in a program is executed.

Appropriate corrections are required.

Claim Objections

6. The following is a quotation of 37 C.F.R § 1.75 (d)(1):

> The claim or claims must conform to the invention as set forth in the remainder of the specification and terms and phrases in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description.

7. Claims 1, 4 and 10 are objected to because of the following informalities:

In Claim 1, Page 10, Lines 12-13, "the value inside the instruction counter and the cycle counter is read to evaluate execution performance" appears to be incorrect and it appears that it should be "the values inside the instruction counter and the cycle counter are read and execution performance is evaluated".

In Claim 1, Page 10, Lines 18-19, "the value inside the instruction counter and the cycle counter is read to evaluate execution performance" appears to be incorrect and it appears that it should be "the values inside the instruction counter and the cycle counter are read and execution performance is evaluated".

In Claim 4, Page 12, Lines 5-6, "initializing the counting by the instruction counter such that the cycle counter increments by one when a timing pulse is traversed" appears to be incorrect and it appears that it should be "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle".

In Claim 10, Page 15, Lines 5-6, "initializing the counting by the instruction counter such that the cycle counter increments by one when a timing pulse is traversed" appears to be incorrect and it appears that it should be "initializing the counting by the

cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle".

Appropriate corrections are required.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the first paragraph of 35 U.S.C. §112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. Claims 1- 11 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

9.1 Claim 1 states in part, "initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed" and "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". The specification does not describe what the initial values used for initializing the counting by the instruction counter and the cycle counter. It also does not describe why the initializing the counting by the instruction counter has to be done when the

microprocessor is in the normal operating mode and initializing the counting by the cycle counter has to be done when the microprocessor is in the emulation mode.

Claim 1 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value". The specification does not describe what an upper value for the instruction counter is and what an upper value for the cycle counter is and how they are selected. It also does not describe how the upper values are loaded into the counters and when.

Claim 1 states in part, "the value inside the instruction counter and the cycle counter is read to evaluate execution performance". It is not clear if the execution performance is actually evaluated or not.

Claim 1 states in step 4," initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed, wherein the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate execution performance", and is step 5, "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle, wherein the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate execution performance". Therefore when the instruction counter is counting in step 4, the cycle counter is not counting. Then when the cycle counter is counting in step 5, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the

two different counters. The values read from the counters at the end of step 4 cannot be used to evaluate the performance. The values read from the counters at the end of step 5 cannot be used to evaluate the performance.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point". How can the program be executed to a definite point, if the microprocessor is in the emulation mode? Are the counter values being incremented in the emulation mode or normal operating mode? When is the definite point in the program set?

9.2 Claim 3 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter and the initialization of counting by the cycle counter are not done as in Claim 1 when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

9.3 Claim 6 states in part, "when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode" and "when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode". The specification does not describe what an upper limit for the instruction counter is and what an upper limit for the cycle counter is and how they are selected. It also does not describe how the upper limits are loaded into the counters and when.

Claim 6 states in element 2, "an instruction counter for counting up by one whenever an instruction is executed, and when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out" and in element 3, "a cycle counter for counting up by one whenever one cycle of timing pulse is traversed, and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out". It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter and the initialization of counting by the cycle counter are not done as in Claim 1. It is also not clear when the microprocessor goes into normal operating mode.

Therefore when the instruction counter is counting in element 2, the cycle counter is not counting. Then when the cycle counter is counting in element 3, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the two different counters. The values read from the counters at the element 2 cannot be used to evaluate the

performance. The values read from the counters at the element 3 cannot be used to evaluate the performance.

9.4 Claim 7 states in part, "initializing the counting by either the instruction counter such that the instruction counter increments by one when an instruction is executed, or by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". Therefore this requires that only one of the two counters be initialized so it can count. However, the performance cannot be evaluated by using only one counter.

The specification does not describe what the initial values used for initializing the counting by the instruction counter and the cycle counter.

Claim 7 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is initialized". The specification does not describe what an upper value for the instruction counter is and what an upper value for the cycle counter is and how they are selected. It also does not describe how the upper values are loaded into the counters and when.

Claim 7 states in part, "reading the value inside the instruction counter and the cycle counter to evaluate execution performance". It is not clear if the execution performance is actually evaluated or not.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode

when the program is executed to a definite point". How can the program be executed to a definite point, if the microprocessor is in the emulation mode? Are the counter values being incremented in the emulation mode or normal operating mode? When is the definite point in the program set?

9.5 Claim 9 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter or the initialization of counting by the cycle counter are not done as in Claim 7 when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

Claims rejected but not specifically addressed are rejected based on their dependency on rejected claims.

10. Claim 1-11 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

10.1 Claim 1 states in part, "initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed" and "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". The specification does not describe what the initial values used for initializing the counting by the instruction counter and the cycle counter. It also does not describe why the initializing the counting by the instruction counter has to be done when the microprocessor is in the normal operating mode and initializing the counting by the cycle counter has to be done when the microprocessor is in the emulation mode.

Claim 1 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value". The specification does not describe what an upper value for the instruction counter is and what an upper value for the cycle counter is and how they are selected. It also does not describe how the upper values are loaded into the counters and when.

Claim 1 states in part, "the value inside the instruction counter and the cycle counter is read to evaluate execution performance". It is not clear if the execution performance is actually evaluated or not.

Claim 1 states in step 4," initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed, wherein the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate

execution performance", and is step 5, "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle, wherein the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate execution performance". Therefore when the instruction counter is counting in step 4, the cycle counter is not counting. Then when the cycle counter is counting in step 5, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the two different counters. The values read from the counters at the end of step 4 cannot be used to evaluate the performance. The values read from the counters at the end of step 5 cannot be used to evaluate the performance.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point". How can the program be executed to a definite point, if the microprocessor is in the emulation mode? Are the counter values being incremented in the emulation mode or normal operating mode? When is the definite point in the program set?

10.2 Claim 3 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter and the initialization of counting by the cycle counter are not done as in Claim 1 when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

10.3 Claim 6 states in part, "when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode" and "when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode". The specification does not describe what an upper limit for the instruction counter is and what an upper limit for the cycle counter is and how they are selected. It also does not describe how the upper limits are loaded into the counters and when.

Claim 6 states in element 2, "an instruction counter for counting up by one whenever an instruction is executed, and when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out" and in element 3, "a cycle counter for counting up by one whenever one cycle of timing pulse is traversed, and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out". It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter and the initialization of counting by the

cycle counter are not done as in Claim 1. It is also not clear when the microprocessor goes into normal operating mode.

Therefore when the instruction counter is counting in element 2, the cycle counter is not counting. Then when the cycle counter is counting in element 3, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the two different counters. The values read from the counters at the element 2 cannot be used to evaluate the performance. The values read from the counters at the element 3 cannot be used to evaluate the performance.

10.4 Claim 7 states in part, "initializing the counting by either the instruction counter such that the instruction counter increments by one when an instruction is executed, or by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". Therefore this requires that only one of the two counters be initialized so it can count. However, the performance cannot be evaluated by using only one counter.

The specification does not describe what the initial values used for initializing the counting by the instruction counter and the cycle counter.

Claim 7 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is initialized". The specification does not describe what an upper value for the instruction counter is and what an upper value for the cycle counter is and

how they are selected. It also does not describe how the upper values are loaded into the counters and when.

Claim 7 states in part, "reading the value inside the instruction counter and the cycle counter to evaluate execution performance". It is not clear if the execution performance is actually evaluated or not.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point". How can the program be executed to a definite point, if the microprocessor is in the emulation mode? Are the counter values being incremented in the emulation mode or normal operating mode? When is the definite point in the program set?

10.5 Claim 9 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter or the initialization of counting by the cycle counter are not done as in Claim 7 when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

Claims rejected but not specifically addressed are rejected based on their dependency on rejected claims.

11. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

12. Claims 1-11 are rejected under 35 U.S.C. 112, second paragraph, as being non-operational.

12.1 Claim 1 states in part, "initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed" and "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". The specification does not describe why the initializing the counting by the instruction counter has to be done when the microprocessor is in the normal operating mode and initializing the counting by the cycle counter has to be done when the microprocessor is in the emulation mode.

Claim 1 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value". The specification does not describe how the upper values are loaded into the counters and when.

Claim 1 states in step 4," initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed, wherein the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate execution performance", and is step 5, "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle, wherein the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value, the value inside the instruction counter and the cycle counter is read to evaluate execution performance". Therefore when the instruction counter is counting in step 4, the cycle counter is not counting. Then when the cycle counter is counting in step 5, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the two different counters. The values read from the counters at the end of step 4 cannot be used to evaluate the performance. The values read from the counters at the end of step 5 cannot be used to evaluate the performance.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point". It is not clear if the program can be executed to a definite point, if the microprocessor is in the emulation mode. It is not clear when the definite point in the program set.

12.2 Claim 3 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear if a starting point for setting the instruction counter and the cycle counter to zero is required. It also not clear if the initialization of the counting by the instruction counter and the initialization of counting by the cycle counter are required or not when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

12.3 Claim 6 states in part, "when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode" and "when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode". It is not clear how the upper limits are loaded into the counters and when.

Claim 6 states in element 2, "an instruction counter for counting up by one whenever an instruction is executed, and when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out" and in element 3, "a cycle counter for counting up by one whenever one cycle of timing pulse is traversed, and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode, the values within the instruction counter and the cycle counter are read out". It is not clear what is the starting point for setting the instruction counter and the cycle counter to zero. It also not clear why the initialization of the counting by the instruction counter and the initialization of counting by the

cycle counter are not done as in Claim 1. It is also not clear when the microprocessor goes into normal operating mode.

Therefore when the instruction counter is counting in element 2, the cycle counter is not counting. Then when the cycle counter is counting in element 3, the instruction counter is not counting. Therefore, the counting is done on different parts of the program by the two different counters. The values read from the counters at the element 2 cannot be used to evaluate the performance. The values read from the counters at the element 3 cannot be used to evaluate the performance.

12.4 Claim 7 states in part, "initializing the counting by either the instruction counter such that the instruction counter increments by one when an instruction is executed, or by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle". Therefore this requires that only one of the two counters be initialized so it can count. However, the performance cannot be evaluated by using only one counter.

Claim 7 states in part, "the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized" and "the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is initialized". The specification does not describe how the upper values are loaded into the counters and when.

Step 5 states, at the end, "then the microprocessor is triggered into the circuit emulation mode again". Then step 6 states, "triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point". It is not clear if the program can be executed

to a definite point, if the microprocessor is in the emulation mode. It is not clear when the definite point in the program set.

12.5 Claim 9 states in part, "setting up an assessment point at an instruction where execution speed is required;

triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution;

reading out the value inside the instruction counter and the cycle counter".

The specification does not state when an assessment point is set. It is not clear if a starting point for setting the instruction counter and the cycle counter to zero is required. It also not clear if the initialization of the counting by the instruction counter and the initialization of counting by the cycle counter are required or not when the assessment point is set. It is also not clear if the microprocessor goes into normal operating mode or not.

Claims rejected but not specifically addressed are rejected based on their dependency on rejected claims.

Claim Interpretations

13. In claim 4, Page 12, Lines 5-6 and claim 10, Page 15, Lines 5-6, "initializing the counting by the instruction counter such that the cycle counter increments by one when a timing

pulse is traversed" is interpreted as "initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle".

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

15. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

16. Claims 1, 6 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Berc et al.** (U.S. Patent 6,112,317) in view of **Killian et al.** (U.S. Patent 6,477,683).

16.1 **Berc et al.** teaches processor performance counter for sampling the execution frequency of individual instructions. Specifically as per claim 1, **Berc et al.** teaches a method of

determining the performance of a microprocessor, wherein the performance of a program having a plurality of instructions is assessed (Abstract, L3-5; CL1, L43-50; CL1, L54-56); comprising the steps of:

resetting an instruction counter and a cycle counter to zero (CL1, L43-50; CL3, L14-19; CL7, L11-14);

initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed (Abstract, L5-8; CL1, L43-50; CL3, L14-19; CL7, L31-33); the value inside the instruction counter and the cycle counter is read to evaluate execution performance (CL3, L21-22);

initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle (CL7, L11-14; CL1, L43-50); the value inside the instruction counter and the cycle counter is read to evaluate execution performance (CL3, L21-22);

reading out the value inside the instruction counter and the cycle counter(CL3, L21-22); and

evaluating microprocessor performance (CL4, L51-53; CL5, L4-8).

Berc et al. does not expressly teach microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode. **Killian et al.** teaches microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state

of the microprocessor including all program visible registers or memory locations (CL13, L15-19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode. The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

Berc et al. does not expressly teach triggering the microprocessor into the normal operating mode and executing the program. **Killian et al.** teaches triggering the microprocessor into the normal operating mode and executing the program (CL13, L19-23), because that allows control of execution of the processor using the debugger and the capability of the on-chip debug mode (CL13, L30-35). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the normal operating mode and executing the program. The artisan would have been motivated because that would allow control of execution of the processor using the debugger and the capability of the on-chip debug mode.

Berc et al. teaches that the microprocessor is triggered to generate an interrupt to an interrupt handler when the instruction counter reaches an upper value; and the microprocessor is triggered to generate an interrupt to an interrupt handler when the cycle counter reaches an upper value (Abstract, L8-10; CL3, L19-20). **Berc et al.** does not expressly teach that the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value; and the microprocessor is triggered into the circuit emulation mode when the

cycle counter reaches an upper value. **Killian et al.** teaches that the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value; and the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state of the microprocessor including all program visible registers or memory locations (CL13, L15-19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included the microprocessor being triggered into the circuit emulation mode when the instruction counter reaches an upper value; and the microprocessor being triggered into the circuit emulation mode when the cycle counter reaches an upper value. The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

Berc et al. does not expressly teach triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point. The artisan would have been motivated because that

would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

16.2 As per claim 6, **Berc et al.** teaches a device for determining the performance of a microprocessor execution (Abstract, L3-5; CL1, L43-50; CL1, L54-56); comprising:

an instruction counter for counting up by one whenever an instruction is executed (Abstract, L5-8; CL1, L43-50; CL3, L14-19; CL7, L31-33); the values within the instruction counter and the cycle counter are read out (CL3, L21-22);

initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle (CL7, L11-14; CL1, L43-50); the values within the instruction counter and the cycle counter are read out (CL3, L21-22); and

microprocessor performance is evaluated by dividing the value inside the cycle counter by the value inside the instruction counter (CL4, L51-53; CL5, L4-8).

Berc et al. does not expressly teach a microprocessor capable of operating in a circuit emulation mode and a normal operating mode. **Killian et al.** teaches a microprocessor capable of operating in a circuit emulation mode and a normal operating mode (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state of the microprocessor including all program visible registers or memory locations (CL13, L15-19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the device of **Berc et al.** with the device of **Killian et al.** that included a microprocessor capable of operating in a circuit emulation mode and a normal operating mode.

The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

Berc et al. teaches that the microprocessor is triggered to generate an interrupt to an interrupt handler when the instruction counter reaches an upper value; and the microprocessor is triggered to generate an interrupt to an interrupt handler when the cycle counter reaches an upper value (Abstract, L8-10; CL3, L19-20). **Berc et al.** does not expressly teach that when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode; and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode. **Killian et al.** teaches that when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode; and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state of the microprocessor including all program visible registers or memory locations (CL13, L15-19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the device of **Berc et al.** with the device of **Killian et al.** that included when the instruction counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode; and when the cycle counter counts to an upper limit, the microprocessor is triggered into the circuit emulation mode. The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

16.3 As per claim 7, **Berc et al.** teaches a method of determining the performance of a microprocessor, wherein the performance of a program having a plurality of instructions is assessed (Abstract, L3-5; CL1, L43-50; CL1, L54-56); comprising the steps of:

resetting an instruction counter and a cycle counter to zero (CL1, L43-50; CL3, L14-19; CL7, L11-14);

initializing the counting by either the instruction counter such that the instruction counter increments by one when an instruction is executed (Abstract, L5-8; CL1, L43-50; CL3, L14-19; CL7, L31-33); or by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle (CL7, L11-14; CL1, L43-50);

reading the value inside the instruction counter and the cycle counter to evaluate execution performance (CL3, L21-22);

reading out the value inside the instruction counter and the cycle counter(CL3, L21-22); and

evaluating microprocessor performance (CL4, L51-53; CL5, L4-8).

Berc et al. does not expressly teach microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode. **Killian et al.** teaches microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state of the microprocessor including all program visible registers or memory locations (CL13, L15-

19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included microprocessor having a circuit emulation mode and a normal operating mode; and triggering the microprocessor into the circuit emulation mode. The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

Berc et al. does not expressly teach triggering the microprocessor into the normal operating mode and executing the program. **Killian et al.** teaches triggering the microprocessor into the normal operating mode and executing the program (CL13, L19-23), because that allows control of execution of the processor using the debugger and the capability of the on-chip debug mode (CL13, L30-35). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the normal operating mode and executing the program. The artisan would have been motivated because that would allow control of execution of the processor using the debugger and the capability of the on-chip debug mode.

Berc et al. teaches that the microprocessor is triggered to generate an interrupt to an interrupt handler when the instruction counter reaches an upper value; and the microprocessor is triggered to generate an interrupt to an interrupt handler when the cycle counter reaches an upper value (Abstract, L8-10; CL3, L19-20). **Berc et al.** does not expressly teach that the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized; and the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is

initialized. **Killian et al.** teaches that the microprocessor is triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized; and the microprocessor is triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is initialized (CL4, L34-39; CL13, L14-18), because the emulation mode (on-chip debug mode) allows accessing the internal state of the microprocessor including all program visible registers or memory locations (CL13, L15-19). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included the microprocessor being triggered into the circuit emulation mode when the instruction counter reaches an upper value if the instruction counter is initialized; and the microprocessor being triggered into the circuit emulation mode when the cycle counter reaches an upper value if the cycle counter is initialized. The artisan would have been motivated because the emulation mode (on-chip debug mode) would allow accessing the internal state of the microprocessor including all program visible registers or memory locations.

Berc et al. does not expressly teach triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when

the program is executed to a definite point. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

17. Claims 2 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Berc et al.** (U.S. Patent 6,112,317) in view of **Killian et al.** (U.S. Patent 6,477,683), and further in view of **Doing et al.** (U.S. Patent 6,018,759).

17.1 As per claim 2, **Berc et al.** and **Killian et al.** teach the method of claim 1. **Berc et al.** does not expressly teach triggering the microprocessor into the circuit emulation mode on complete execution of the program; reading out the value inside the instruction counter and the cycle counter; and evaluating microprocessor performance. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when the program is executed to a definite point. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

Doing et al. teaches the microprocessor identifying completion of execution of the program; reading out the value inside the instruction counter and the cycle counter; and evaluating microprocessor performance (CL22, L21-28), because that allows determining the relative performance of target program during the execution (CL22, L21-22). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Doing et al.** that included the microprocessor identifying completion of execution of the program; reading out the value inside the instruction counter and the cycle counter; and evaluating microprocessor performance. The artisan would have been motivated because that would allow determining the relative performance of target program during the execution.

17.2 As per Claim 8, it is rejected based on the same reasoning as Claim 2, supra. Claim 8 is a method claim reciting the same limitations as Claim 2, as taught throughout by **Berc et al., Killian et al.** and **Doing et al.**

18. Claims 3-5 and 9-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Berc et al.** (U.S. Patent 6,112,317) in view of **Killian et al.** (U.S. Patent 6,477,683), and further in view of **Doing et al.** (U.S. Patent 6,018,759) and **Roth et al.** U.S. Patent Application 2002/0078329).

18.1 As per claim 3, **Berc et al., Killian et al.** and **Doing et al.** teach the method of claim 2. **Berc et al.** teaches reading out the value inside the instruction counter and the cycle counter (CL3, L21-22); and

evaluating microprocessor performance (CL4, L51-53; CL5, L4-8).

Berc et al. does not expressly teach setting up an assessment point at an instruction where execution speed is required. **Killian et al.** teaches setting up an assessment point (watch point) at an instruction where execution speed is required (CL31, L36-38), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included setting up an assessment point at an instruction where execution speed is required. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

In addition, **Roth et al.** teaches setting up an assessment point (watch point) at an instruction where execution speed is required (Page 1, Para 0002 and Para 0014), because that allows extracting the state information from the processor at watch points and providing the state information to a control unit (Page 1, Para 0002); and identifying the emulation mode and provide emulation mode operation handling (Page 1, Para 0012). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Roth et al.** that included setting up an assessment point at an instruction where execution speed is required. The artisan would have been motivated because

that would allow extracting the state information from the processor at watch points and providing the state information to a control unit; and identifying the emulation mode and provide emulation mode operation handling.

Berc et al. does not expressly teach triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when the assessment point is encountered during instruction execution. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

18.4 As per claim 4, **Berc et al., Killian et al., Doing et al.** and **Roth et al.** teach the method of claim 2. **Berc et al.** teaches reading out the value inside the instruction counter and the cycle counter (CL3, L21-22);

resetting an instruction counter and a cycle counter to zero (CL1, L43-50; CL3, L14-19; CL7, L11-14);

initializing the counting by the instruction counter such that the instruction counter increments by one when an instruction is executed (Abstract, L5-8; CL1, L43-50; CL3, L14-19; CL7, L31-33);

initializing the counting by the cycle counter such that the cycle counter increments by one when a timing pulse traverses a cycle (CL7, L11-14; CL1, L43-50); and

evaluating microprocessor performance (CL4, L51-53; CL5, L4-8).

Berc et al. does not expressly teach setting up an assessment point at the start and at the end of a series of instructions where execution speed is required. **Killian et al.** teaches setting up an assessment point (watch point) at the start and at the end of a series of instructions where execution speed is required (CL31, L36-38), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included setting up an assessment point at the start and at the end of a series of instructions where execution speed is required. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

In addition, **Roth et al.** teaches setting up an assessment point (watch point) at the start and at the end of a series of instructions where execution speed is required (Page 1, Para 0002 and Para 0014), because that allows extracting the state information from the processor at watch points and providing the state information to a control unit (Page 1, Para 0002); and identifying

the emulation mode and provide emulation mode operation handling (Page 1, Para 0012). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Roth et al.** that included setting up an assessment point at the start and at the end of a series of instructions where execution speed is required. The artisan would have been motivated because that would allow extracting the state information from the processor at watch points and providing the state information to a control unit; and identifying the emulation mode and provide emulation mode operation handling.

Berc et al. does not expressly teach triggering the microprocessor into the circuit emulation mode when the start assessment point is encountered during instruction execution. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the start assessment point is encountered during instruction execution (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when the start assessment point is encountered during instruction execution. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

Berc et al. does not expressly teach triggering the microprocessor into the normal operating mode and executing the program. **Killian et al.** teaches triggering the microprocessor into the normal operating mode and executing the program (CL13, L19-23), because that allows

control of execution of the processor using the debugger and the capability of the on-chip debug mode (CL13, L30-35). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the normal operating mode and executing the program. The artisan would have been motivated because that would allow control of execution of the processor using the debugger and the capability of the on-chip debug mode.

Berc et al. does not expressly teach triggering the microprocessor into the circuit emulation mode when the ending assessment point is encountered during instruction execution. **Killian et al.** teaches triggering the microprocessor into the circuit emulation mode when the ending assessment point is encountered during instruction execution (CL11, L66-67; CL29, L47-50; CL31, L36-38 and L46-49), because that allows sampling the counters of the processor at those points (CL31, L63-66); and comparing the states at the breakpoint (CL31, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Berc et al.** with the method of **Killian et al.** that included triggering the microprocessor into the circuit emulation mode when the ending assessment point is encountered during instruction execution. The artisan would have been motivated because that would allow sampling the counters of the processor at those points; and comparing the states at the breakpoint.

Per claim 5: **Berc et al.** teaches dividing the value inside cycle counter by the value inside the instruction counter (CL4, L51-53; CL5, L4-8).

18.5 As per Claims 9-11, these are rejected based on the same reasoning as Claims 3-5, supra. Claims 9-11 are method claims reciting the same limitations as Claims 3-5, as taught throughout by **Berc et al., Killian et al., Doing et al.** and **Roth et al.**

Conclusion

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dr. Kandasamy Thangavelu whose telephone number is 571-272-3717. The examiner can normally be reached on Monday through Friday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska, can be reached on 571-272-3716. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-9600.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

K. Thangavelu
Art Unit 2123
March 5, 2005